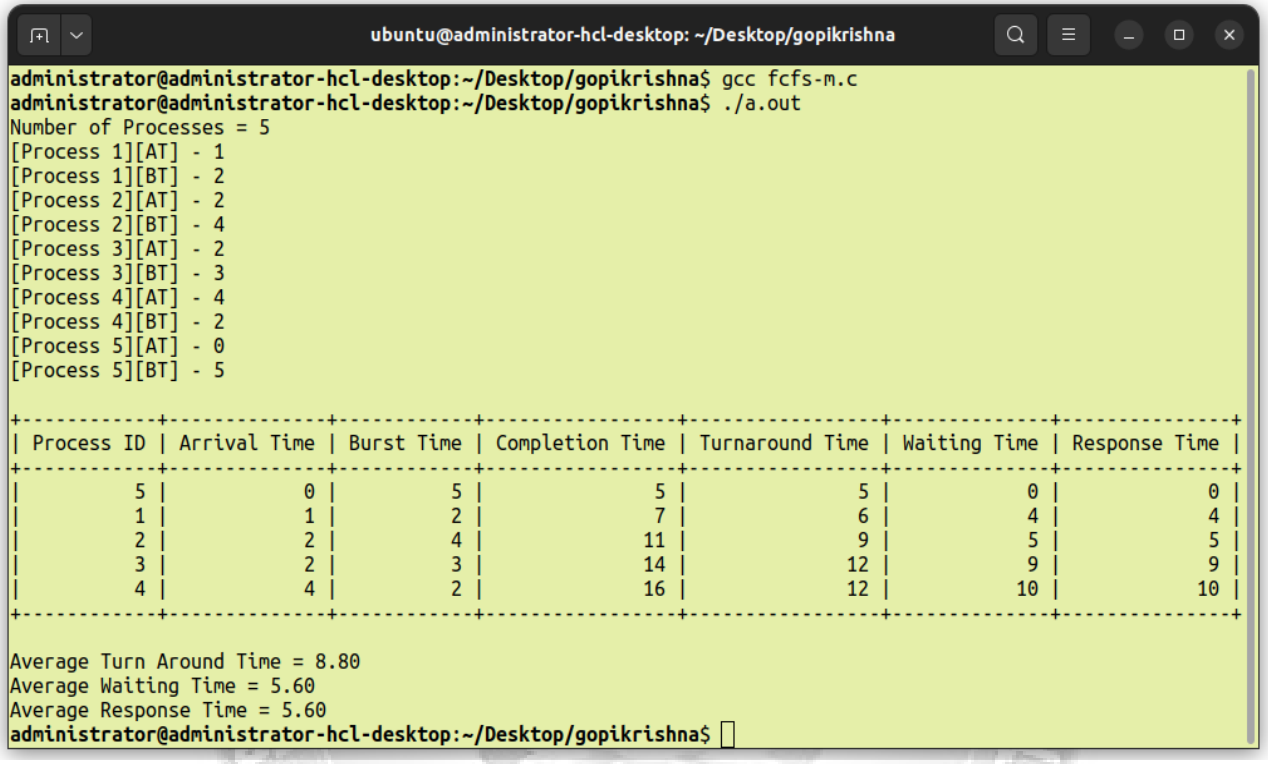
PROGRAM CODE

```
#include <stdio.h>

void calculateTime(int n,int process[], int arrival[], int burst[], int completion[], int turn_around[],
int waiting[], int response[])
{
   int i,total_waiting = 0,total_turnaround = 0,total_response = 0,current_time = 0;

   for (i = 0; i < n; i++)
   {
      if (current_time > arrival[i])
      {
         completion[i] = current_time + burst[i];
      }
      else
      {
         completion[i] = arrival[i] + burst[i];
      }

      turn_around[i] = completion[i] - arrival[i];

      waiting[i] = turn_around[i] - burst[i];
      if (waiting[i] < 0)
      {
         waiting[i] = 0;
      }

      response[i] = waiting[i];

      current_time = completion[i];

      total_waiting += waiting[i];
      total_turnaround += turn_around[i];
      total_response += response[i];
   }

   float avg_waiting = (float)total_waiting / n;
   float avg_turnaround = (float)total_turnaround / n;
   float avg_response = (float)total_response / n;

   printf("\n+------------+--------------+------------+-----------------+-----------------+--------------
+---------------+\n");
   printf("| Process ID | Arrival Time | Burst Time | Completion Time | Turnaround Time | Waiting
Time | Response Time |\n");
   printf("+------------+--------------+------------+-----------------+-----------------+--------------
+---------------+\n");
   for (i = 0; i < n; i++)
   {
      printf("| %10d | %12d |%11d | %15d |  %14d |%13d |%14d |\n", process[i], arrival[i], burst[i],
completion[i], turn_around[i],
      waiting[i], response[i]);
```

```
    }
    printf("+------------+-------------+-----------+----------------+----------------+-------------
+---------------+\n");
    printf("\nAverage Turn Around Time = %.2f\n", avg_turnaround);
    printf("Average Waiting Time = %.2f\n", avg_waiting);
    printf("Average Response Time = %.2f\n", avg_response);
}

void sortDataByArrival(int n,int process[], int arrival[], int burst[])
{
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = 0; j < n - i - 1; j++)
        {
            if (arrival[j] > arrival[j + 1])
            {
                int temp_process = process[j];
                process[j] = process[j + 1];
                process[j + 1] = temp_process;

                int temp_arrival = arrival[j];
                arrival[j] = arrival[j + 1];
                arrival[j + 1] = temp_arrival;

                int temp_burst = burst[j];
                burst[j] = burst[j + 1];
                burst[j + 1] = temp_burst;
            }
        }
    }
}

int main()
{
    int n;
    printf("Number of Processes = ");
    scanf("%d", &n);

    int process[n],arrival[n],burst[n],completion[n],turn_around[n],waiting[n],response[n];

    for (int i = 0; i < n; i++)
    {
        process[i]=i+1;
        printf("[Process %d][AT] - ",i+1);
        scanf("%d", &arrival[i]);
        printf("[Process %d][BT] - ",i+1);
        scanf("%d", &burst[i]);
    }

    sortDataByArrival(n,process,arrival,burst);
    calculateTime(n, process, arrival, burst, completion, turn_around, waiting, response);
```

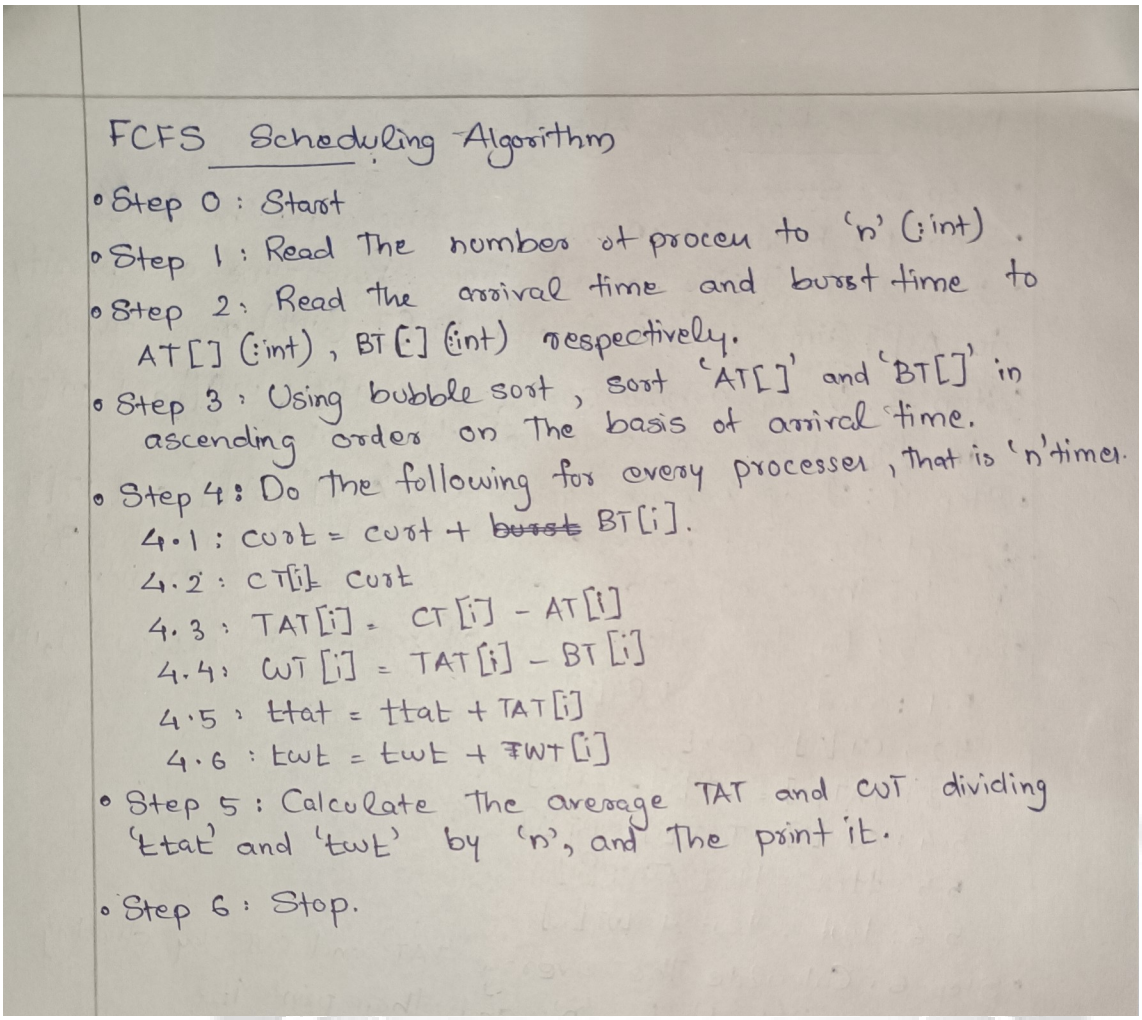
```
    return 0;
}
```

OUTPUT



```
administrator@administrator-hcl-desktop:~/Desktop/gopikrishna$ gcc fcfs-m.c
administrator@administrator-hcl-desktop:~/Desktop/gopikrishna$ ./a.out
Number of Processes = 5
[Process 1][AT] - 1
[Process 1][BT] - 2
[Process 2][AT] - 2
[Process 2][BT] - 4
[Process 3][AT] - 2
[Process 3][BT] - 3
[Process 4][AT] - 4
[Process 4][BT] - 2
[Process 5][AT] - 0
[Process 5][BT] - 5

+------------+--------------+------------+-----------------+-----------------+--------------+---------------+
| Process ID | Arrival Time | Burst Time | Completion Time | Turnaround Time | Waiting Time | Response Time |
+------------+--------------+------------+-----------------+-----------------+--------------+---------------+
|          5 |            0 |          5 |               5 |               5 |            0 |             0 |
|          1 |            1 |          2 |               7 |               6 |            4 |             4 |
|          2 |            2 |          4 |              11 |               9 |            5 |             5 |
|          3 |            2 |          3 |              14 |              12 |            9 |             9 |
|          4 |            4 |          2 |              16 |              12 |           10 |            10 |
+------------+--------------+------------+-----------------+-----------------+--------------+---------------+

Average Turn Around Time = 8.80
Average Waiting Time = 5.60
Average Response Time = 5.60
administrator@administrator-hcl-desktop:~/Desktop/gopikrishna$
```

ALGORITHM



FCFS Scheduling Algorithm

- Step 0 : Start
- Step 1 : Read the number of proces to 'n' (:int) .
- Step 2 : Read the arrival time and burst time to AT[] (:int) , BT[:] (int) respectively.
- Step 3 : Using bubble sort , sort 'AT[]' and 'BT[]' in ascending order on the basis of arrival time.
- Step 4 : Do the following for every processes , that is 'n' times.
  - 4.1 : cust = cust + ~~burst~~ BT[i].
  - 4.2 : CT[i] cust
  - 4.3 : TAT[i] = CT[i] - AT[i]
  - 4.4 : CWT[i] = TAT[i] - BT[i]
  - 4.5 : ttat = ttat + TAT[i]
  - 4.6 : twt = twt + WT[i]
- Step 5 : Calculate the average TAT and CWT dividing 'ttat' and 'twt' by 'n', and the print it.
- Step 6 : Stop.